Performance Investigation and Optimization of IEEE802.15.4 for Industrial Wireless Sensor Networks

MOHSIN HAMEED, HENNING TRSEK, OLAF GRAESER AND JUERGEN JASPERNEITE

Presented By: Aniket Shah

Outline

- Introduction
- Related Work
- Engineering Aspects
- Performance Evaluation
- Limitations
- GTS scheduling and optimization
- Conclusion
- Review

Introduction

- Use of Guaranteed Time Slots (GTS) as a medium access control mechanism for real time data transmission in WSN
- GTS limited by number of nodes usage and scalability
- Introduction of Earliest Due Date GTS Allocation (EDDGTSA); a scheduling algorithm

Introduction

- IEEE 802.15.4 has become a standard for LR-WPAN
- Features:
  - Communication Area < 10m (POS)
  - Transfer Rate: 20,40, 100, 250 kbps
  - Provides GTS
- Two network configuration nodes:
  - Beacon enabled
  - Non beacon enabled

Introduction

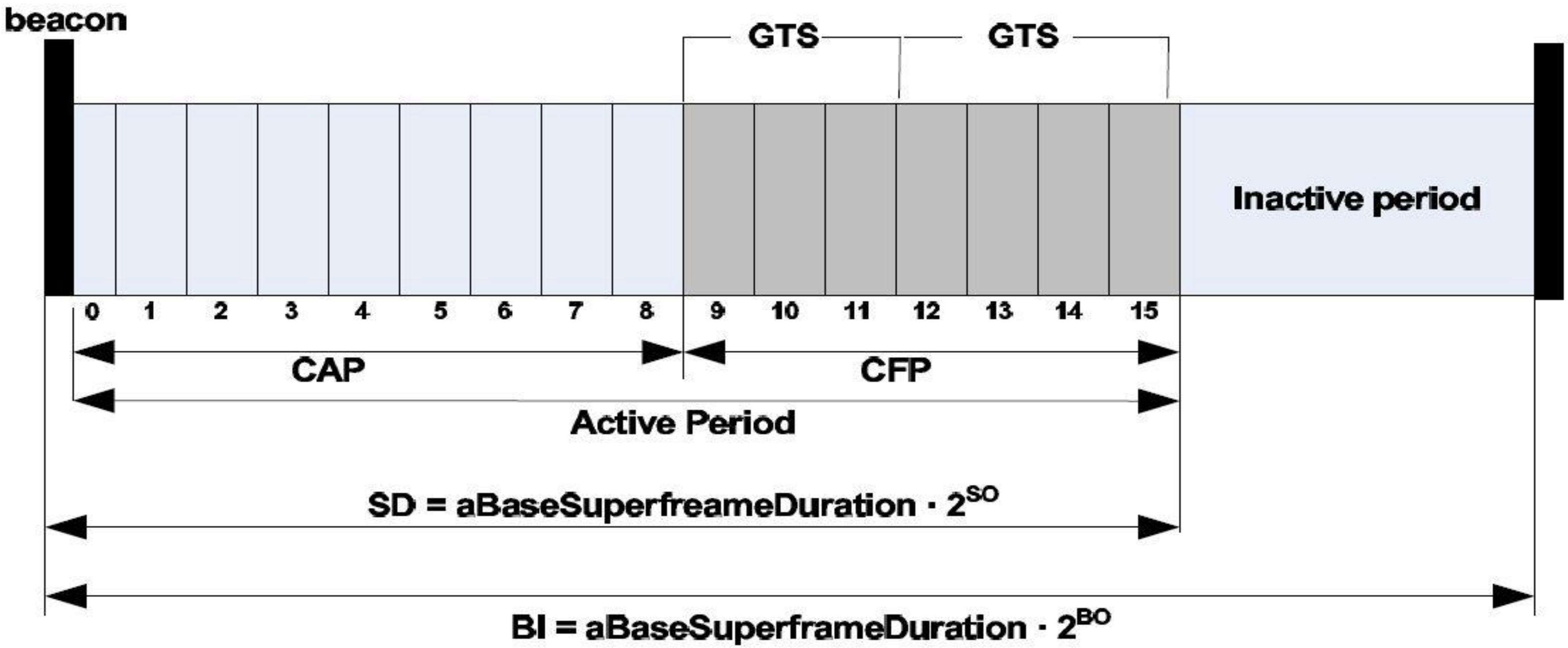
- PHY and MAC layers defined
- PHY layer:
  - Use DSSS to spread across all frequency bands
  - ISM frequency bands used as shown below

FREQUENCY (MHz)	NO. OF CHANNELS	DATA RATES (kbps) / MODULATION
868 – 868.6	1	20 / BPSK, 100 / O-QPSK, 250 / ASK/O-QPSK
902 – 298	10	40 / BPSK, 250 / ASK/O-QPSK
2400 – 2483.5	16	250 / O-QPSK

Introduction

- MAC layer:
  - Bacon management
  - Channel access
  - GTS management
  - Frame validation
  - Frame delivery acknowledgements
  - Association and Dis-association
- This paper focuses on beacon enabled mode operating at 2.4 GHz ISM frequency and data rate of 250 kbps

Super-frame Structure



[Fig 1]

Super-frame Structure

- Two parameter: Beacon Order (***BO***) and Superframe Order (***SO***) where $0 \le SO \le BO \le 14$
- If ***SO*** = 15; superframe is not active & if ***BO*** = 15; superframe doesn't exist and Non beacon enabled mode used
- ***Superframe Duration (SD) = aBaseSuperframeDuration . 2^{SO} ;***
  Beacon Interval (BI) = aBaseSuperframeDuration . 2^{BO} ***Eq. (1) and (2)***
- ***aBaseSuperframeDuration = aBaseSlotDuration . aNumSuperframeSlots Eq. (3)***
- ***aBaseSlotDuration =*** No. of symbols forming superframe slot & ***aNumSuperframeSlots =*** No. of slots in any superframe

Super-frame Structure

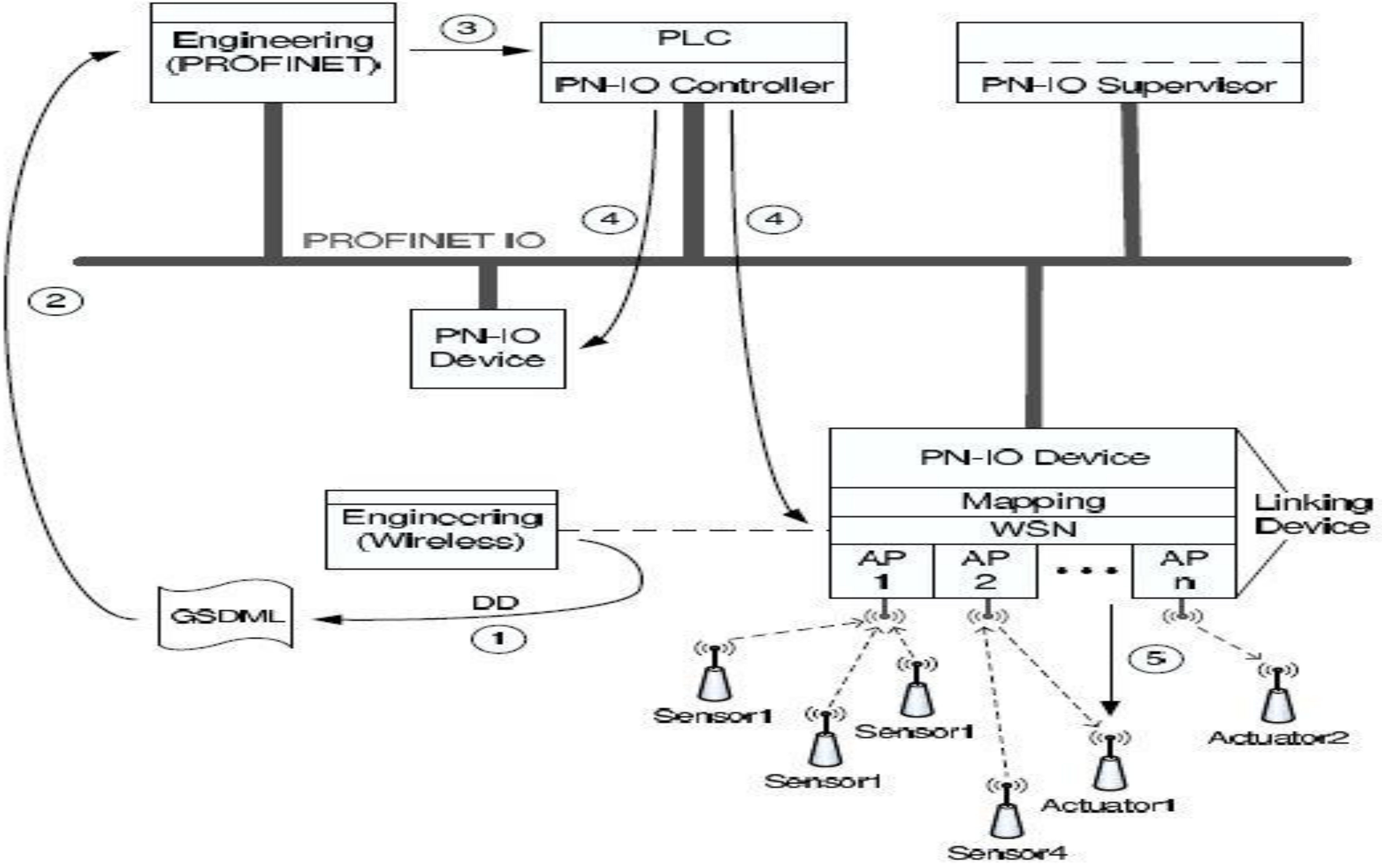
- Contention Access Period (CAP) uses CSMA mechanism
- Contention Free Period (CFP) uses GTS; can be activated by request from node
- Minimum CAP length = 440 symbols

Related Work

- Previous evaluations on security and energy efficiency
- IEEE 802.15.4 in factory automation with delay consideration
- GTS behavior analysis with respect to delay and throughput
- GTS scheduling schemes are assessed

Engineering Aspects

- Industrial Automation is based on static offline configuration that impacts WSN handling
- Use of Industrial Ethernet Standard PROFINET using a generic markup language GSDML
- GSDML file transferred to PROFINET IO tool and then to the controller to configure all devices
- GSDML file helps with mapping by providing WSN configuration.
- **Problem** : No dynamic behavior leading to static network configuration; **Solution** : Scheduling after startup phase

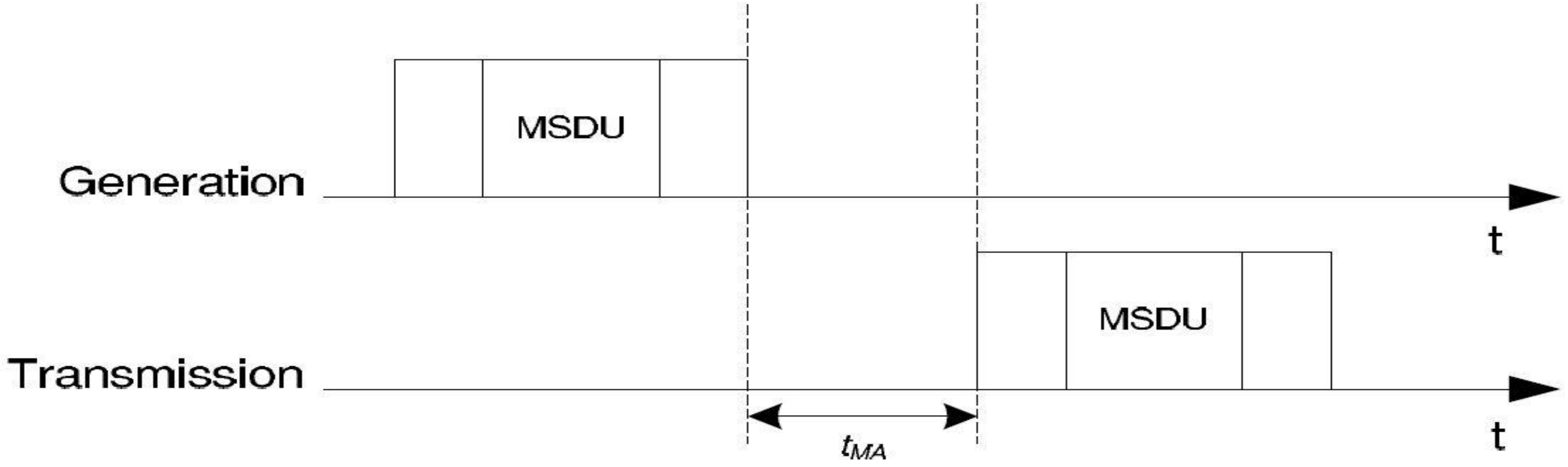


Engineering of Industrial Automation System

[Fig 2]

Performance Evaluation

- OPNET simulation model developed as per Koubba for 802.15.4 [8]
- Main metric for performance evaluation is Medium Access Delay
- Medium Access Delay = time interval between frame generation and actual medium access of frame

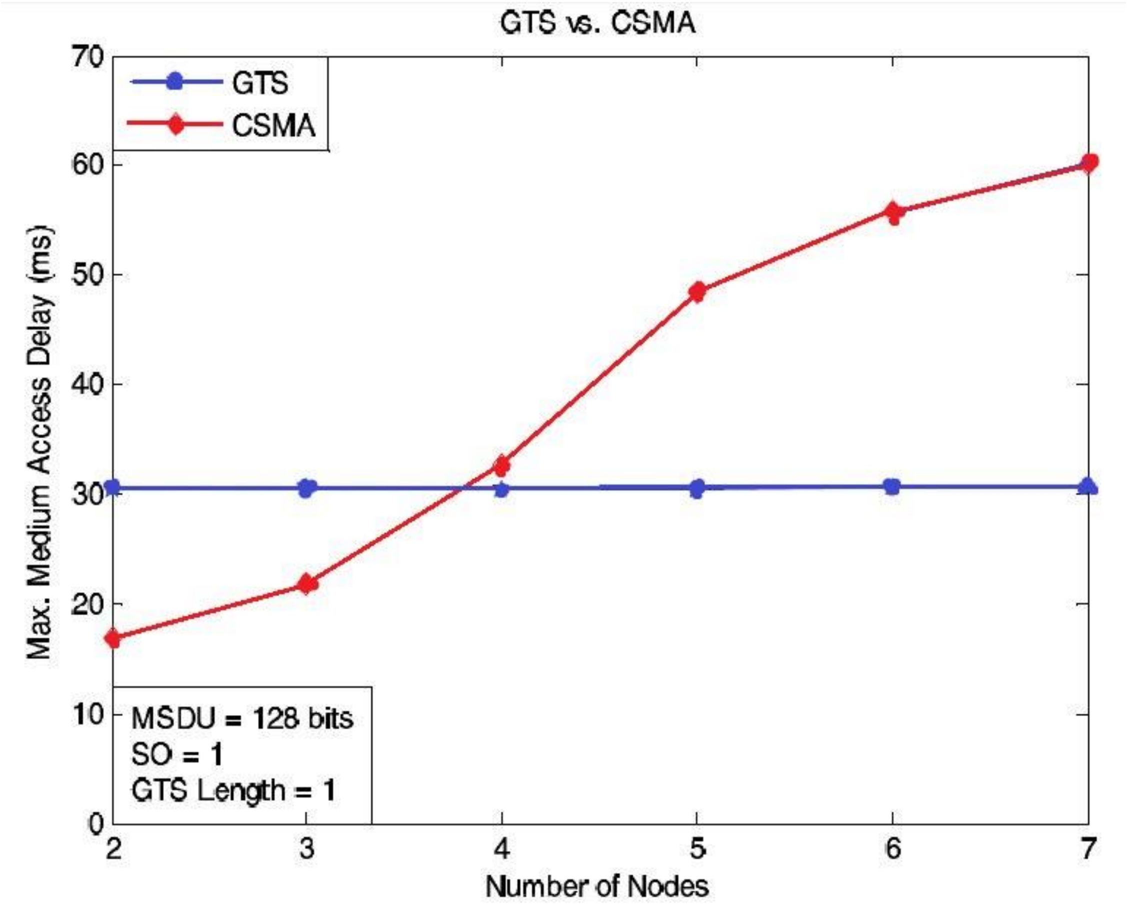


[Fig 3]

Performance Evaluation

- For CSMA, t_{MA} depends on node back-off time, for GTS, t_{MA} depends on GTS length, SO and payload size
- Scenario 1: Delay vs Number of Nodes
  - Interval time = 1s
  - ***SO*** = ***BO*** = 1
  - MSDU size = 128 bits
- Result: GTS performs better than CSMA as number of nodes increase
- Reason: CSMA delay increases steeply due to more collisions on channel due to increased network load

GTS vs CSMA delay comparison

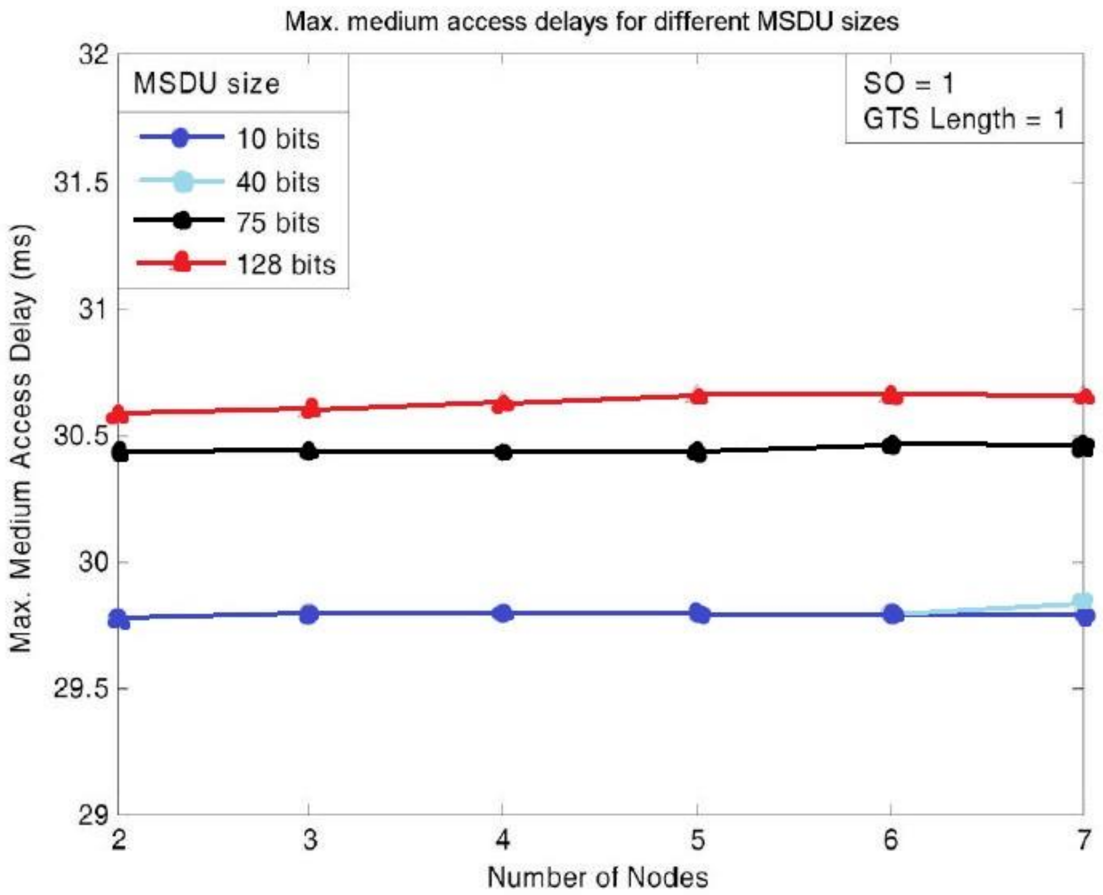


[Fig 4]

Performance Evaluation

- Scenario 2: Max Delay in GTS for different MSDU sizes for varying no. of nodes
  - SO = BO = 1
  - GTS length = 1
  - MSDU size = 10, 40, 75, 128 bits
- Result: For MSDU size of 40 bits or lower, the medium access delay < 30ms while for MSDU size > 40 bits, 30 ms > medium access delay > 31 ms
- Observations: Payload size and Number of nodes do not significantly affect medium access delay

Max Delay vs Number of nodes & MSDU size

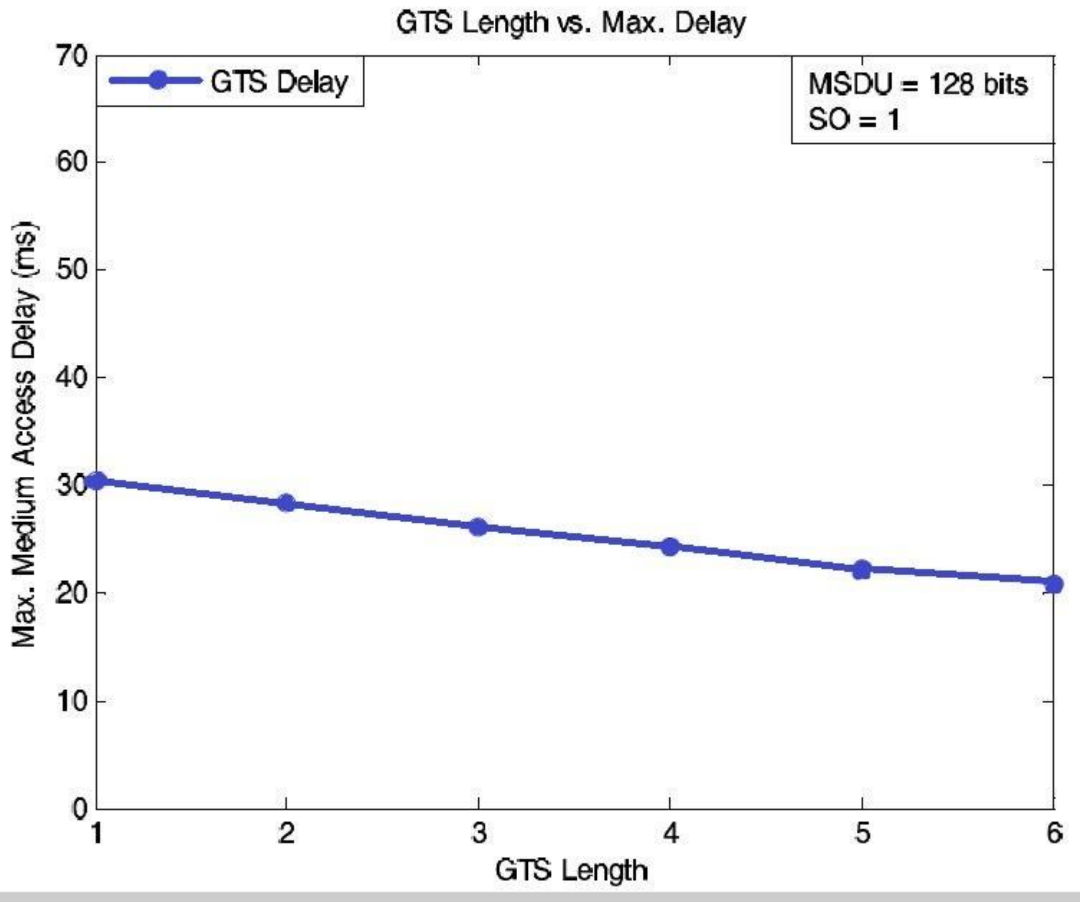


[Fig 5]

Performance Evaluation

- Scenario 3: Effect of GTS length on max delay for 2 nodes
  - MSDU size = 128 nits
  - SO = BO = 1
- Result: Increase in GTS length significantly reduces max delay
- Observation: Increase in GTS length decreases number of nodes used

GTS length vs Max Delay



[Fig 6]

Limitations

- Max 7 GTSs in one superframe
- Exclusive dedication of every GTS to its respective node; Thus, max 7 nodes at a time can be supported
- Scalability for large scale industrial application using WSN
- **Solution:** Optimized GTS Scheduling scheme

Optimized GTS Scheduling

- Introduction of Earliest Due Date GTS Allocation (EDDGTSA), an optimized scheduling algorithm
- Basic concept is to schedule nodes based on their maximum allowed delays
- Input for EDDGTSA:

Table 1. Input of the EDDGTSA algorithm

Name	*Description*
listOfNodes	List of currently accepted nodes
newNode	New node (with normDelay and reqTS)
normDelay	Normalized delay for newNode
reqTS	Required time slots for newNode

Optimized GTS Scheduling

- All nodes send max delay to PAN coordinator
- Max delay is normalized as a multiple of Beacon Interval and superframe cycle, given by normDelay
- **normDelay = maxDelay / BI**
- EDDGTSA requests list of all nodes as argument to handle node sorting
- Table of nodes created with each row consisting of nodes with same normalized delay

Algorithm 1 EDDGTSA algorithm

```
procedure EDDGTSA(listOfNodes, newNode)
    listOfAllNodes.add(newNode)
            ▷ newNode contains normDelay and reqTS
    table :=createTableNodeDelay(listOfNodes)
            ▷ rows: normDelay, columns: nodes
    superframes :=createArrayOfSuperframes()
    sf_number :=1
    while table ≠ ∅ do
        sf := superframes[sf_number]
        while sf. hasUnassignedSlots() do
            sf.assignNodeMinAllowedDelay(table)
            table.removeNodeMinAllowedDelay()
        superframes.add(sf)
        for all row_i ∈ table do
            if sf_number mod i = 0 then
                if row_i ≠ ∅ then
                        ▷ Time requirements not fulfilled
                    return DENIED
        if table = ∅ then
            return superframes          ▷ SUCCESS
        for all row_i ∈ table do
            if sf_number mod i = 0 then
                table.fillRow(i, listOfNodes)
        sf_number := sf_number + 1
    return b                            ▷ The gcd is b
```

Optimized GTS Scheduling

- Algorithm creates a chain of superframes; first 7 slots of first superframe assigned to nodes with smallest normDelay
- Assigned nodes removed from table ....(lines 11,12)
- Steps for filling superframe, deleting nodes from table and refiling specific nodes of table are repeated until table is completely filled
- When complete table is empty, algorithm stops because scheduling task is finished

Optimized GTS Scheduling

- Number of rows r_i empty after assembling superframe SF_j are determined by the equation as shown

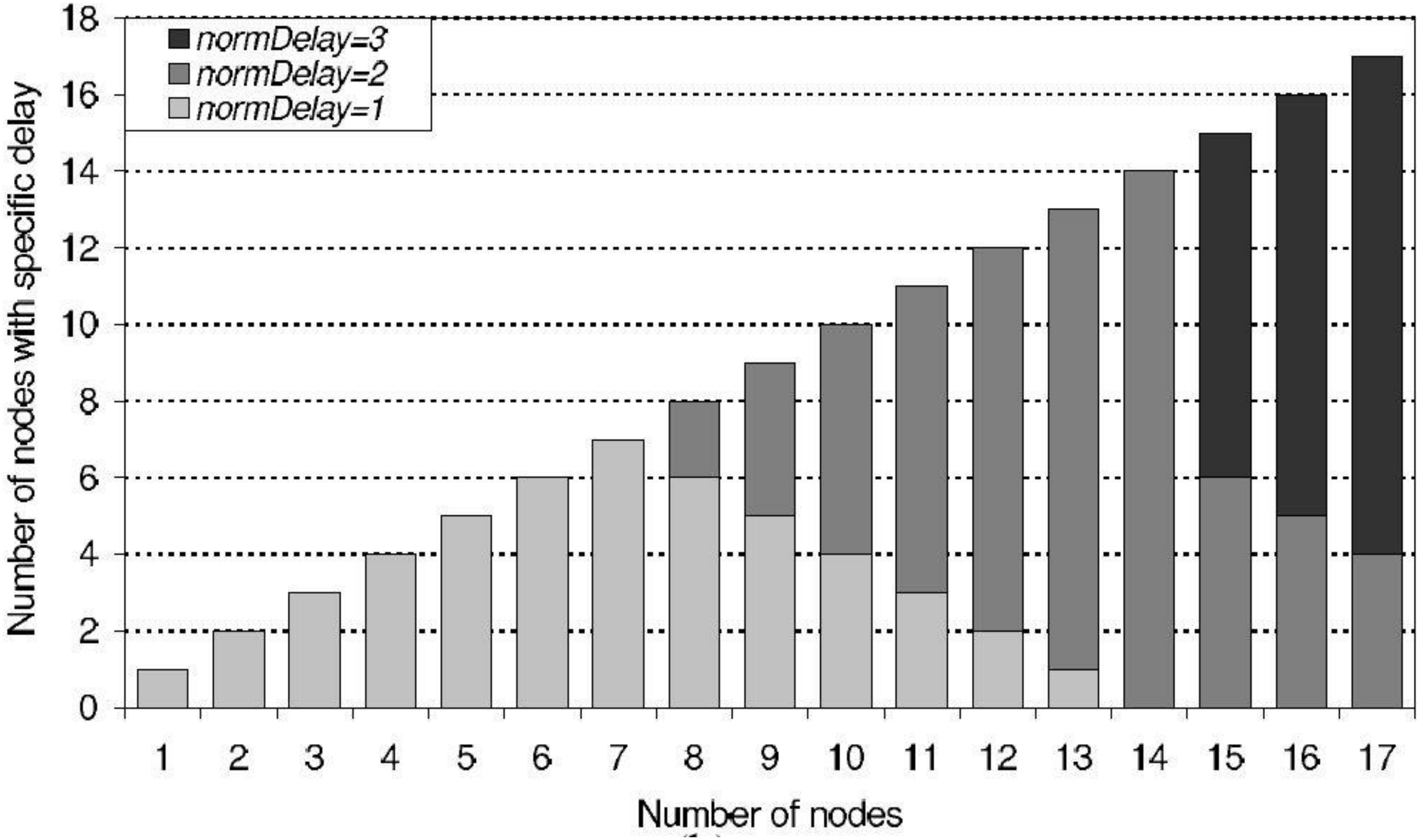
- r_i = { must be checked for emptiness and refilled; if j mod i = 0
  . { must neither be checked nor refilled; otherwise

- Worst Case Scenarios:
  - When each of the ***n*** nodes requires 7 slots, max allowed delay ≥ ***n* or more** cycles, the algorithm requires ***n*** superframes resulting in ***n*** iterations of the ***while*** loop
  - When each node has a different max allowed delay, table consists of ***n*** nodes resulting in the ***n*** iterations of the ***for all*** loops

Optimized GTS Scheduling

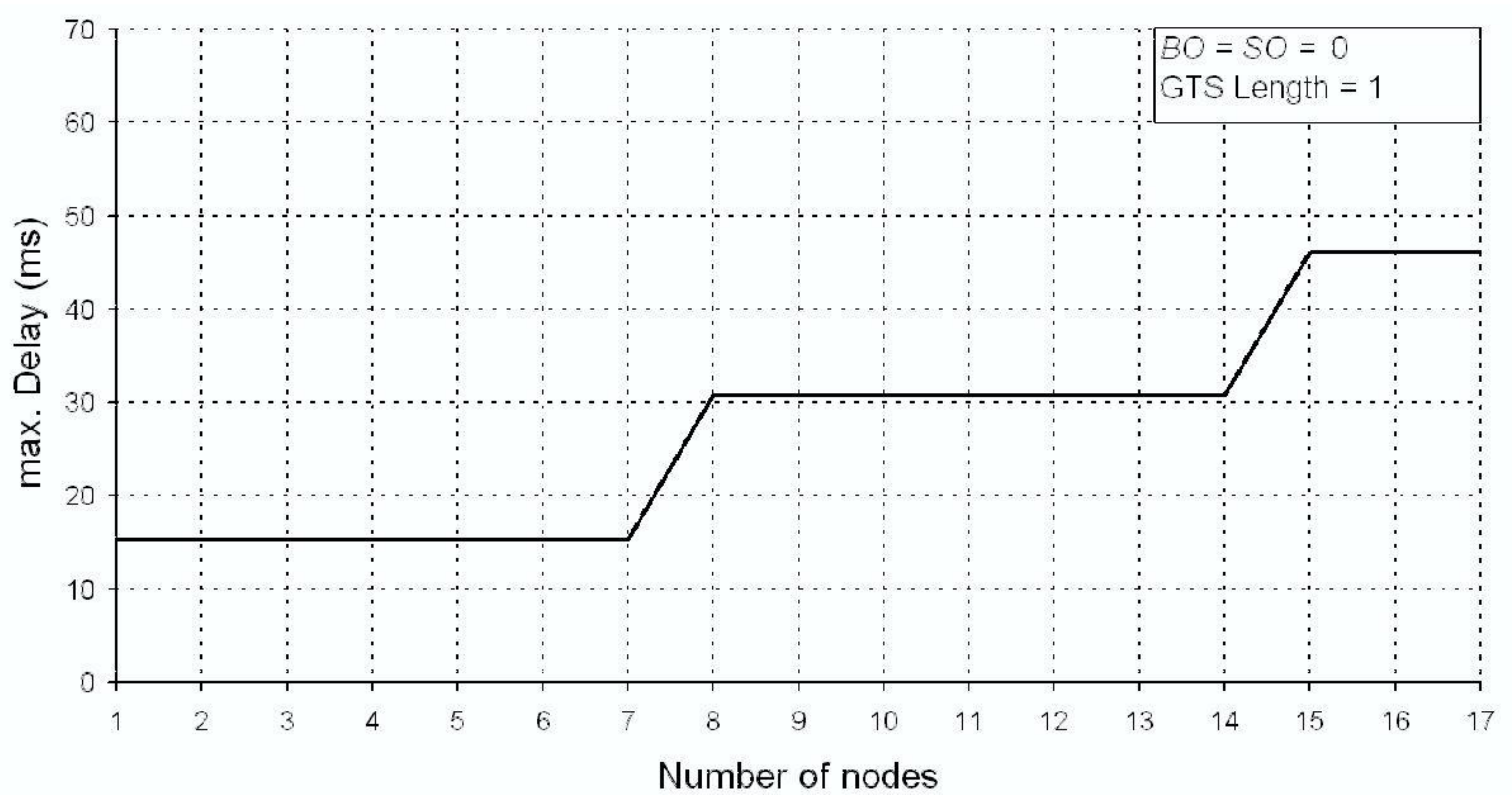
- Upper bound for algorithm is given by $O(n^2)$
- Assumptions:
  - Effect of Collisions were disregarded for analytic calculations
  - No packets were lost during transmission
- Reason: GTS mechanism provides a contention free period which results in zero collisions
- Results of the algorithm performance for ***normDelay*** are shown below. It shows the number of nodes connected to the coordinator and the requirements for different scenarios

No. of Nodes wrt Requested Max Delay



[Fig 7 (b)]

Max Allowed Delay vs No. of Nodes



Condition of experiment: *Requirement of every node is identical*

[Fig 8]

Conclusion

- GTS outperformed CSMA; maintained its bounds while CSMA fulfilled requirements only with fewer nodes
- GTS mechanisms has its limitations that can be overcome by using EDDGTSA
- EDDGTSA allows multiple nodes to share same GTS time slots in different superframes based on their max allowed delays
- EDDGTSA works reasonably well in industrial WSNs and should be deployed more
- Future work: Detailed simulation study of proposed algorithm for further refinement and implementation on an evaluation platform

Review

- Authors cover an important topic with regards to IEEE 802.15.4 communication, i.e. scheduling of time slots wrt number of nodes
- Provide convincing, readable results for their experimentation
- Could have provided more detail on the OPNET simulation model and maybe evaluated on a few more metrics
- As a reviewer, I wouldn’t accept the paper as I feel there hasn’t been enough experimentation done

Questions